Hexagonal Global Parameterization of Arbitrary Surfaces

Matthias Nieser, Jonathan Palacios, Konrad Polthier, and Eugene Zhang

Abstract—In this paper we introduce hexagonal global parameterizations, a new type of periodic global parameterization that respects six-fold rotational symmetries (6-RoSy). Such parameterizations enable the tiling of surfaces with regular hexagonal texture and geometry patterns and can be used to generate high-quality triangular remeshing. To construct a hexagonal parameterization given a surface, we provide an automatic technique to generate a 6-RoSy field that respects directional and singularity features in the surface. This is achieved by applying the trace-and-deviator decomposition to the curvature tensor, which allows us to identify regions of appropriate directional constraints. We also introduce a technique for automatically merging and cancelling singularities. This field will then be used to generate a hexagonal global parameterization by adapting the framework of QuadCover parameterization. In particular, we formulate the energy terms needed to solve for a hexagonal parameterization. We demonstrate the usefulness of our geometry-aware global parameterization with applications such as surface tiling with regular textures and geometry patterns and triangular remeshing.

Index Terms—Surface parameterization, rotational symmetry, hexagonal tiling, triangular remeshing, surface sampling.

1 INTRODUCTION

In this paper we introduce hexagonal global parameterizations, a new type of periodic global parameterizations that are ideal for tiling surfaces with patterns of six-fold rotational symmetries, i.e., 6-RoSys [1]. Such hexagonal patterns possess some rather interesting properties. First, they are one of the only two classes of rotationally-invariant patterns that can seamlessly tile a plane (the other being quadrangular patterns). Second, hexagonal tilings provide optimal approximation to circle packing. These properties have been linked to the wide appearances of hexagonal patterns in nature, such as honeycombs, insect eyes, fish eggs, and snow and water crystals, as well as in man-made objects such as floor tiling, carpet patterns, and architectural decorations. Figure 1 shows some examples of hexagonal patterns in nature and man-made objects.

Tiling a surface with regular texture and geometry patterns is an important yet challenging problem in pattern synthesis. Methods based on some local parameterization of the surface often lead to visible breakup of the patterns along seams, i.e., where the surface is cut open during parameterization. Periodic global parameterizations can alleviate this problem when the translational and rotational discontinuity in the parameterization is compatible with the tiling pattern in the input texture and geometry. For example, a quadrangular global parameterization is designed to be compatible with an input texture of 4-RoSy (Figure 2 (a)). On the other hand, it is incompatible with 6-RoSy patterns (hexagonal) (Figure 2 (b)). In contrast, a hexagonal global parameterization is compatible with 6-RoSy patterns (Figure 2 (c)). The hexagonal global parameterizations we introduce in this paper are designed to facilitate the tiling of surfaces with hexagonal texture and geometry patterns.

Another motivation of our work is triangular remeshing, in which an input triangular mesh is used to generate a new triangular mesh that approximates the same underlying surface with improved quality. In triangular remeshing, it is often desirable to have all the triangles in the mesh being equilateral and of uniform sizes. Furthermore, the irregular vertices in the mesh can lead to difficulties in subsequently applications. With these conditions, we can reformulate the problem of triangular remeshing into the problem of triangular tiling of the underlying surface, with the irregular vertices in the mesh corresponding to the singularities in the tiling. Note that the dual of a triangular tiling is a hexagonal tiling. Consequently, triangular remeshing can be accomplished by computing a hexagonal global parameterization of the input mesh. This provides an alternative to existing approaches that perform Delaunay triangulation or centroidal Voronoi tessellation. A benefit of using hexagonal parameterizations in remeshing is the ability to explicitly control the number, type, and location of irregular vertices in the new mesh.

The automatic generation of a hexagonal parameterization from an input surface has a number of challenges. First, unlike quadrangular parameterizations whose parameter lines are parallel to either the major or the minor principal
curvature directions, in hexagonal parameterizations only one of the two directions can be used at each point on the surface. One must decide which direction to choose, and how to propagate such choices from a relatively small set of points to the whole surface to maintain the smoothness of the resulting parameterization. Second, existing techniques to explicitly control the singularities in the parameterization are user-driven (including quadrangular parameterizations), and it is not an easy task to provide automatic control over the number and location of such singularities. Third, the energy terms developed for quadrangular parameterizations are not appropriate for hexagonal parameterizations, and to the best of our knowledge no such energy terms have been formulated.

To address these challenges, we present a two-step pipeline. First, a 6-RoSy field on the surface is automatically generated based on the curvature tensor of the surface. Our field generation technique applies the trace-and-deviator decomposition to the curvature tensor, which allows us to select appropriate sparse directional constraints from either the major or minor eigenvectors in the curvature tensor. This results in remeshings whose edges are aligned with the bending of the surface, so that the shape can be more aptly represented with relatively few triangles or hexagons. Further, we introduce a method for automatically clustering singularities, inspired by mesh decimation techniques. The smoothness of the field, along with the reduced number of singularities lead to highly regular meshes with a relatively small number of irregular vertices. Such meshes are desirable for subdivision surface applications [2]. In the second step of our pipeline, we solve a sparse linear system based on an energy term that we have developed specifically for hexagonal parameterizations. The resulting parameters can then be used to generate triangular meshes free of T-junctions. We also point out that the singularities in the hexagonal parameterization need to occur on the set of Eisenstein integers, which is different than the quadrangular case, where the singularities occur on the Gauss integers. This leads to the parameterization method that we refer to as HexCover.

In summary, our contributions in this paper are as follows:

1) We introduce the hexagonal global parameterization and demonstrate its uses with applications such as triangular remeshing and surface tiling with regular texture and geometry patterns.
2) We present the first technique to construct a hexagonal global parameterization given an input surface.
3) We point out the importance of geometry-aware 6-RoSy fields for hexagonal global parameterization.
4) We propose an automated pipeline for generating geometry-aware 6-RoSy fields. As part of the pipeline, we develop a metric for locating parabolic features on surfaces based on tensor decomposition as well as a way of automatically clustering singularities.
5) We formulate the energy term for generating a hexagonal global parameterization from a 6-RoSy field.

The remainder of this paper is organized as follows. We first cover related work in relevant research areas in Section 2. Next, we describe our pipeline for generating a geometry-aware 6-RoSy field given an input surface in Section 3, and our parameterization technique in Section 4. In Section 5, we demonstrate the usefulness of our techniques with applications in triangular remeshing and surface tiling with regular texture and geometry patterns. Lastly, we close in Section 6 with our conclusions.

2 RELATED WORK

Surface Parameterization: Surface parameterization is a well-explored research area. We will not attempt a complete review of the literature but instead refer the reader to the surveys by Floater and Hormann [3] and Hormann et al. [4]. A very early surface parameterization method is the Tutte’s barycentric graph embedding [5]. Tutte’s embeddings are combinatorial in nature and do not capture the geometry of
Fig. 3. We introduce hexagonal global parameterizations (an example on the dragon is shown in the upper-left), which can be used to perform regular texture and geometry pattern synthesis with patterns of six-fold symmetries (upper-right and lower-left) as well as geometry-aware triangular remeshing (lower-right).

the surface. Early global parameterization methods focus on conformal parameterizations [6], [7], [8], which are aimed at angle preservation at the cost of length distortion.

To reduce length distortion, Kharevych et al. [9] use cone singularities for conformal parameterization, which relax the constraint of a flat parameter domain at few isolated points. Such singularities have proven essential for high quality parameterizations and have been used in other parameterization schemes as well.

Tong et al. [10] use singularities at the vertices of a hand-picked quadrilateral meta layout on the surface. The patches of the meta layout are then parameterized by solving for a global harmonic one-form. Dong et al. [11] use a similar idea for parameterization but create the quadrilateral meta layout automatically from the Morse-Smale complex of Eigenfunctions of the mesh Laplacian.

Ray et al. [12] parameterize surfaces of arbitrary genus with periodic potential functions guided by two orthogonal input vector fields, or a 4-RoSy field. This leads to a continuous parameterization except in the vicinity of singularities on the surface. These singular regions are detected and reparameterized afterwards. The QuadCover algorithm [13] builds upon this idea by using the input 4-RoSy field to construct a four-fold covering space on which it becomes a vector field. This covering space and vector field is then used to generate a global parameterization with no T-junctions. Our algorithm to generate a parameterization from a 6-RoSy field is an adaptation of the QuadCover method, with a different energy formulation specified designed for hexagonal global parameterization.

Bommes et al. [14] propose a method similar to the aforementioned techniques, but formulate the parameterization as the solution to a mixed-integer system of equations. They also add constraints that force parameter lines to capture sharp edges.

**Field Processing:** Much work has been done on the subject of vector and tensor (1- and 2-RoSy) field anal-
ysis. To review all of this work is beyond the scope of this paper; here we refer to only the most relevant work. Helman and Hesselink [15] propose a method of vector field visualization based on topological analysis and provide methods of extracting vector field singularities and separatrices. Topological analysis techniques for symmetric second-order tensor fields are later introduced in [16]. In the context of vector field design, numerous systems have been developed for the purpose of vector field, most of which have been for specific graphics applications such as texture synthesis [17], fluid simulation [20], and vector field visualization [21], [22]. Fisher et al. [23] propose a vector field design system based on discrete one-forms. Note that the above systems do not employ any methods of topological analysis, and do not extract singularities and separatrices. Systems providing topological analysis include [24], [25] and [26]. The last has also been extended to design tensor fields [27], [28]. In contrast, relatively little work has been done on N-RoSy fields when \( N > 2 \). Hertzmann and Zorin [29] utilize cross or 4-RoSy fields in their work on non-photorealistic pen-and-ink sketching, and provide a method for smoothing such fields. Ray et al. [30] extend the surface vector field representation proposed in [25] into a design system for N-RoSy fields of arbitrary \( N \). Palacios and Zhang [1] propose an N-RoSy design system that allows initialization using design elements as well as topological editing of existing fields. They also provide analysis techniques for the purpose of locating both singularities and separatrices, and a visualization technique in [31]. Lai et al. [32] propose a design method based on a Riemannian metric, that gives the user control over the number and locations of singularities. Their system also allows for mixed N-RoSy fields, with different values of \( N \) in different regions of the mesh. However, this method is based on user design while we focus on automatic and geometry-aware generation. Bommes et al. [14] offer a method of producing a smooth 4-RoSy field from sparse constraints, formulated as a mixed-integer problem.

Ray et al. [33] propose a framework to generate an N-RoSy field that follows the natural directions in the surface and has a reduced number of singularities which tends to fall into natural locations. In this paper, we make use of this framework but automatically generate the input constraints, which relieves the user from labor-intensive manual design.

3 GEOMETRY-AWARE 6-RoSy FIELD GENERATION

In this section, we describe our pipeline for generating a geometry-aware 6-RoSy field \( F \) given an input surface \( S \). This field will then be used to guide the parameterization stage of our algorithm (Section 4).

We first review some relevant properties of 6-RoSy fields [1], [30]. An \( N \)-RoSy field \( F \) has a set of \( N \) directions at each point \( p \) in the domain of the field:

\[
F(p) = \{ R_N^i v(p) \}, \quad i \in \{0, \ldots, N-1\}
\]

where the vector \( v(p) = R(p) \begin{pmatrix} \cos \theta(p) \\ \sin \theta(p) \end{pmatrix} \) is one of the \( N \) directions, and \( R_N^i \) is the linear operator that rotates a given vector by \( 2\pi i/N \) in the corresponding tangent plane. A singularity is a point \( p_0 \) such that \( \rho(p_0) = 0 \) and \( \theta(p_0) \) is undefined; \( p_0 \) is isolated if the value of \( \rho \neq 0 \) for all points in a sufficiently small neighborhood of \( p_0 \), except at \( p_0 \). An isolated N-RoSy singularity can be measured by its index, which is defined in terms of the Gauss map [1] and has an index of \( \pi I \), where \( I \in \mathbb{Z} \). A singularity \( p_0 \) is of first-order if \( I = \pm 1 \). When \( |I| > 1 \), \( p_0 \) is referred to as a higher-order singularity. A higher-order singularity with an index of \( \frac{I}{N} \) can be realized by merging \( I \) first-order singularities.

Requirements and Pipeline: There are a number of goals that we wish to achieve with our automatic field generation.

First, we wish to control the number, location, and type of the singularities in the field. When performing quad-ruangular and triangular remeshing using periodic global parameterizations, the singularities in the guiding 4- or 6-RoSy field correspond to the irregular vertices in the mesh. Such singularities can also lead to the breakup of texture and geometry patterns during pattern synthesis on surfaces. Consequently, the ability to control the number, location, and type of singularities in the field can improve quality of the remeshes and surface tilings.

Second, the field needs to be smooth, or distortion can occur in the resulting parameterizations that have undesirable effects for triangular remeshing and surface tiling.

Third, we need the parameter lines in the parameterization to be aligned with the feature lines in the surfaces, such as ridge and valley lines (see Figure 4). In addition, it has been documented that having texture directions aligned with the feature lines in the mesh can improve the visual perception of the texture synthesis [34].

Note that these requirements may conflict with each other. For example, excessive reduction of singularities can lead
to high distortion in the field, and an overly-smoothed field may deviate from feature lines. To deal with this we adopt the framework of Ray et al. [33]. Given a set of user-specified constraints and a field curvature, the framework generates a sparse linear system whose solution is the RoSy field that matches the constraints and field curvature in the least square sense. Each constraint represents a desired $N$-RoSy value, i.e., $N$ directions, at a given point. In our case we wish to have our field aligned with principal curvature directions. The user-specified field curvature is a vertex-based function defined on the mesh, whose value at a vertex represents the desired discrete Gauss curvature at this vertex. The field curvature must sum up to $2 \pi \chi(S)$ where $\chi(S)$ is the Euler characteristic of the surface $S$. It allows the user to specify the location and type of singularities in the field. For example, a vertex with a field curvature of $\frac{2 \pi}{N}$, after the system is solved, is likely to have a singularity of index $\frac{k}{N}$ there.

Given a surface with complex geometry and topology, it can be labor intensive to provide all necessary constraints and the field curvature through a lengthy trial-and-error process. Consequently, we automatically generate the directional constraints as well as the field curvature, which is at the core of our algorithm for field generation. Our algorithm consists of two stages. First, we identify a set of directional constraints based on the curvature and solve for an initial 6-RoSy field using these constraints only. Second, we extract all the singularities in the initial field and perform iterative singularity pair clustering until the distance between any singularity pair is above a given threshold. The remaining singularities will be used to create the field curvature, which, along with the initial directional constraints, will be used to generate the final RoSy field. We now describe each of these stages in more detail next.

**Automatic Constraint Identification:** To automatically identify directional constraints, we need to answer the questions of where to place constraints and what direction is assigned to each constraint.

Recall that we wish to align the parameter lines with feature lines such as ridges and valleys, i.e., the principal direction in which the least bending occurs. Note that the directions in the 6-RoSy field is the gradient of the parameterization (Section 4). Consequently, we will choose the principal direction that has the most bending, i.e., maximum absolute principal curvature, as one of the directions in the 6-RoSy.

Principal curvature directions are most meaningful in parabolic and hyperbolic regions due to the strong anisotropy there. However, while purely hyperbolic regions possess strong anisotropy, the absolute principal curvatures are nearly indistinguishable, thus making both principal curvature directions an candidate. Moreover, the two bi-sectors between the major and minor principal curvature directions also provide viable choices for the edge directions in hyperbolic regions. Due to the excessive choices of directions in hyperbolic regions and insufficient choice of directions in planar and spherical regions, we only generate directional constraints in parabolic regions.

We make use of a representation of the curvature tensor that readily exposes where on this spectrum of classification any point on a given surface falls. Using the trace-and-deviator decomposition similar to those employed in [27], we find that the curvature tensor $T$ at any point $p \in S$ can be rewritten in the form:

$$ T = \left( \frac{k_1 - k_2}{2} \cos \theta \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix} + \frac{k_1 + k_2}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) $$

$$ = \frac{\rho}{\sqrt{2}} \left( \cos \phi \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix} + \sin \phi \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) $$

(1)

where $k_1$ and $k_2$ are the principal curvatures at $p$, $\rho = \sqrt{k_1^2 + k_2^2}$, $\phi \in [-\pi/2, \pi/2] = \arctan(\frac{k_1 + k_2}{k_1 - k_2})$, and $\theta \in [0, 2\pi)$ is the angular component of the maximum principal direction measured in the local frame at $p$. Note that the first component in the sum is traceless and symmetric, while the second is a multiple of the identity matrix. $T(p)$ can now be classified using $(\rho(p), \phi(p))$, which spans a half plane. There are six special configurations on this half plane. For the remaining five configurations we have $\rho(p) > 0$. Respectively, they correspond to $\phi(p) = \frac{\pi}{2}$ (spherical), $\phi(p) = \frac{\pi}{4}$ (cylindrical), $\phi(p) = 0$ (purely hyperbolic), $\phi(p) = -\frac{\pi}{4}$ (inverted cylindrical), and $\phi(p) = \frac{\pi}{2}$ (inverted spherical). With this representation, we can classify any point $p$ as being planar if $\phi(p)$ is smaller than a given threshold $\delta$, elliptical if $\phi(p) \geq \delta$ and $|\phi(p)| \leq \frac{\pi}{6}$, hyperbolic if $\phi(p) \geq \delta$ and $|\phi(p)| > \frac{\pi}{6}$, and

![Fig. 5. A summary of our surface classification. On the left, we show $\phi \in [-\pi/2, \pi/2]$ color mapped to the [BLUE,RED] arc of the HSV color wheel on the Stanford Bunny; the top shows a continuous mapping and the bottom, a binned classification based on the legend shown on the right. Also shown on the right are the surfaces to which points with given values of $\phi$ are locally similar and the relationships between the principal curvatures for each of these surfaces. The maximum curvature directions are shown in red, and the minimum in blue.](image)
Fig. 6. Here we show a summary of our constraint selection. On the right, we show the Buddha with color mapped to $\rho$. We then select the 35 percent of the surface with the highest values of $\rho$ (middle); here we have colored the constraints based on $\phi$ using the legend in Figure 5. Finally, we select the maximum curvature directions as constraints in the regions where $\phi > 0$ (yellow) and the minimum where $\phi < 0$ (cyan). In the close-up on the left, we can see that constraints selected in this fashion are always orthogonal to the direction in which the surface is bending the most. Also, notice that the nearly adjacent yellow and cyan regions do not conflict via a $\pi/2$ rotation, as they would if we had selected the only one of the curvature directions everywhere.

parabolic otherwise, i.e., $\phi(\mathbf{p}) \geq \delta$ and $\frac{\pi}{8} \leq |\phi(\mathbf{p})| \leq \frac{3\pi}{8}$. Note that this tensor representation is a direct application of the eigenvalue manifold for asymmetric tensor fields [35] to the principal curvature tensor.

Given the classification, we propagate the directions in the parabolic regions into non-parabolic regions (planar, spherical, hyperbolic) using energy minimization, an approach taken in [14]. To accomplish this, we pick the points where $\rho$ (the tensor magnitude) is above certain a threshold $t_\rho$, and label these points as having “strong” curvature (in all of our examples, we have chosen $t_\rho$ so that 35 percent of the area of $S$ is so-labeled). From this set of points, we use only the directions of the parabolic points as constraints; that is, the points for which $\phi \in [-3\pi/8, -\pi/8] \cup [\pi/8, 3\pi/8]$ (Figure 6). Finally, we select the maximum direction $\theta$ as the constraint direction at points where $\phi > 0$ and the minimum direction $\theta + \pi/2$ where $\phi < 0$. Recall that the directions in the output field specify the gradients in our resulting parameterizations, and we wish one the isolines of the parameters to be orthogonal to the direction in which the surface is bending the most. Clearly, the above directions satisfy this requirement (See the shapes on the right side of the right image in Figure 5). Finally, the constraints are used to set up a linear system [33] whose solution gives rise to our initial RoSy field.

For our solver, we use the geometry-aware $N$-RoSy field generation technique proposed by [33], as it allows us to control the level of geometric detail that is reflected by singularities, and also plays a role in the implementation of our singularity clustering technique. This system, based on Discrete Exterior Calculus (DEC) [36], filters (locally averages) the Gauss Curvature $K$ of $S$ to produce $\bar{K}$ and then computes a target field curvature $C'$ using the difference between $K$ and $\bar{K}$. $C'$ is then used to modify the angles by which directions rotate when parallel transported along mesh edges. This compensates for the actual curvature of $S$, and directional fields computed on $S$ under these conditions behave as though $S$ has a Gauss curvature of $\bar{K}$. Since $\bar{K}$ is smoother than $K$, such fields have reduced topological noise, which makes them more suitable for our parameterization algorithm.

**Automatic Singularity Clustering:** Our initial field was obtained from directional constraints only. Consequently, it typically consists of only first-order singularities. Given a surface with rather complex geometry and topology, the number of singularities can be rather large. Furthermore, while the location of the singularities tend to be appropriate (in high curvature regions), many of them form dense clusters. Having singularities in closer proximity can lead to difficulties in the resulting parameterization. This is because the singularities will be constrained to be on a lattice in the parameter space as typically required by most global parameterization methods [13], [14]. Consequently, the smallest distance between any singularity pair will be mapped to a unit in the parameter space. If the smallest distance is too small, the two involved singularities may be mapped to the same point on the lattice, leading to a locally infinite stretching in the parameterization.

To address this, many field generation techniques constrain the number of singularities to be as few as possible [30], but this represents another extreme, where the field directions can become highly distorted in some regions. Furthermore, many of the aforementioned approaches require much user interaction [1], [30], [33], which can be time-consuming for models with complex geometry and topology.

Our goal is to automatically reduce the number of singularities in the field while retaining the locations of the remaining singularities inside high curvature regions. To achieve this we employ the following process.

First, we extract the singularities in the initial RoSy field [30] which we use to build a graph embedded in the surface. The nodes of this graph are the singularities in the field, and the edges representing proximity information between singularity pairs. We refer to this graph as the *singularity graph* $G$. To construct $G$, we compute a Voronoi diagram with the singularities as sites. The dual graph gives rise to the singularity graph [37].

Second, we iteratively perform edge collapse to this graph, which is equivalent to performing singularity pair clustering (merging or cancellation), until the minimal surface distance between any singularity pair is above a given threshold. Every time a singularity pair is clustered, we compute the sum of the singularity indexes and place a singular constraint with the sum as its desired index. Note that we do this even if the sum is zero, i.e., singularity
Fig. 7. A visual summary of our method for automatically clustering singularities. Our initial field has a large number of singularities (a). In (b) we show the singularity graph $G$ constructed from the field in (a: ear, nose). We then perform a series of edge-collapses to get the reduced graph (c); the region $R$ is shown in green. Finally, we resolve the field in $R$ using the nodes of $G$ as singular constraints and the boundary of $R$ as directional constraints. This gives us a reduced field (d) which has fewer singularities. Note the singularities are colored based on the sign of their indexes, not the actual value. In fact, there are higher-order singularities (d) but none in (a).

pair cancellation. The singularity constraint is placed on the path between the two original singularities, closest to the one with the Gaussian curvature of highest magnitude. This is an attempt to keep singularities near the features that caused them to originally appear during initialization and is accomplished by interpolating along the geodesic from $p_0$ to $p_1$ using the value $|K(p_1)|(|K(p_0) + K(p_1)|)$, where $K(p)$ is the Gaussian curvature at $p \in S$. We continue to collapse edges in the order of increasing edge-length on $G$ until no edge of length less than $d_{\text{sing}}$ remains. At the end of this process, we will have generated a set of singularity constraints, i.e., the remaining vertices in the graph, which is then used to update the field in the vicinity of these singularities. In the case of fields generated for remeshing, $d_{\text{sing}}$ can be selected based on the edge-length of the output mesh. We choose $d_{\text{sing}}$ to be $0.1B$ where $B$ is the size of the bounding box for the model. For a visual summary of the algorithm, see Figure 7.

Third, we modify the field curvature based on the singularity constraints. Recall that the field curvature is simply a smoothed version of the discrete Gauss curvature during the generation of the initial field. The singularity constraints, produced in the previous step, consists of a set of vertices in the mesh and a desired singularity index $\tau(p)$ for each such singularity constraint $p$. We modify the field curvature such that it is zero everywhere on the surface except at singularity constraints where the value of the field curvature is $\frac{2\pi}{N}\tau(p)$. Notice that such assignment satisfies that the total desired field curvature is equal to the Euler characteristic of the underlying surface. We now modify the 6-RoSy field by solving the same system used to generate the initial field, with two difference. We do not solve the system everywhere on the surface. Instead, we generate a region $R_{p_0} = \{p | d(p, p_0) < d_{\text{sing}}\}$ for each singularity constraint $p_0$ and solve the system in the union of these disks. That is, the field values are fixed for vertices outside these regions and will serve as the boundary condition when solving the system. The field values for vertices inside the disks are changed. We have found this to be fast and efficient in controlling the location, type, and number of singularities.

We wish to point out that our automatic field generation method can be applied to $N$-RoSy field generation for any $N$, in particular 4-RoSy fields. Figure 8 shows an example generated using our method. The only change in the whole field generation pipeline occurs during automatic
identification of directional constraints. Instead of choosing \( \theta \) or \( \theta + \frac{\pi}{2} \) as one of the six directions for 6-RoSy fields, we choose both for 4-RoSy fields.

4 HexCover Parameterization

We now describe the algorithm behind the second stage of our pipeline. To do so we first define some concepts.

Given a triangular mesh surface \( S \), a periodic global parameterization of \( S \) with respect to an \( N \)-RoSy field is a collection of linear maps \( \alpha_i = (u_i, v_i) \) that map each triangle \( t_i \in S \) onto \( \mathbb{R}^2 \) with the following property. For all pairs of adjacent triangles \( t_i, t_j \), the transition function between them must satisfy:

\[
\alpha_i(p) = R_{ij}^\theta \alpha_j(p) + t_{ij}, \quad \forall p \in t_i \cap t_j,
\]

where \( r_{ij} \in \{0, 1, \ldots, N - 1\} \) and \( t_{ij} \) are the rotational and translational discontinuities, respectively. Recall that \( R_{ij}^\theta \) is the linear operator that rotates a vector by \( \frac{2\pi r_{ij}}{N} \).

For quadrangular global parameterizations \( t_{ij} \) are required to lie on the set of Gauss integers, defined as \( \{(a, b) | a, b \in \mathbb{Z}\} \). This constraint eliminates seams in surfacing and T-junctions in remeshing. Recent approaches generate a global parameterization by solving an energy minimization problem. Käbläer et al. [13] define the energy formulation, which is quadratic, based on the concept of covering spaces from algebraic topology. They solve the system using a conjugate gradient solver. Bommes et al. [14] employ a similar energy term but solve it using a different solver.

In this paper we wish to compute a hexagonal global parameterization from a 6-RoSy field using the approach of [13], [14], i.e., turning the parameterization problem into an energy minimization problem. To do we need to answer two questions. First, what is the energy like for a hexagonal global parameterization? Second, what is the set on which \( t_{ij} \) and the singularities are situated? Both questions can answered using the idea of covering spaces.

Energy for Hexagonal Parameterizations: In the interest of saving space we do not review the definition of covering spaces. Instead, we provide some practical descriptions which are needed to understand our algorithm.

Given a mesh surface \( S \) and a 6-RoSy field \( F \) defined on \( S \), we construct a six-fold covering \( S' \) of \( (S, F) \) along with two vector fields \( F'_u \) and \( F'_v \). Every triangle \( t \) in \( S \) will have six corresponding triangles in \( S' \): \( t_0, \ldots, t_5 \) (Figure 9). \( F'_u \) is a vector field on \( S' \) that distributes the six vectors of the 6-RoSy field onto the six copies, i.e., \( F'_u(t_i) = R_{ij}^\theta F_u(t) \) where \( f(t) \) is one of the six directions in the 6-RoSy in \( t \). Beside \( F'_u \), we also define \( F'_v = R_{ij}^\theta F'_u \) in \( S' \). The two vector fields are mutually perpendicular, and they are the gradients of the parameters \( u \) and \( v \), respectively, i.e., \( \nabla u = F'_u \) and \( \nabla v = F'_v \). Together they provide the energy term required for energy minimization:

\[
E(u', v') := \int_{M'} (\|\nabla u' - F_u\|^2 + \|\nabla v' - F_u\|^2) dA
\]

Given two adjacent triangles \( s \) and \( t \) in \( S \), their corresponding layers in \( S' \) are combinatorially glued together by some matching \( d_{st} \in \{0, 1, 2, 3, 4, 5\} \), which connects the sheets \( s_t \) with \( t_{t+d_{st}} \). The matching describes which of the 6-RoSy vectors in \( s \) and \( t \) are paired. While any matching assignment \( d_{st} \) is valid combinatorially, in our implementation we compute \( d_{st} \) that matches the vectors from neighboring triangles to be most aligned. This is achieved by setting \( i + d_{st} = \arg \min_k F_u(s_t) \cdot F_u(t_k) \). Once it is established that \( s_t \) and \( t_j \) are neighbors in \( S' \), we have \( s_{t+p} \) and \( t_{j+p} \) \((1 \leq p \leq 5)\) are neighbors in \( S' \) as well.

Notice that to compute \( u \) in \( S \) requires to choose a vector from the 6-RoSy inside each triangle. This can lead to not only a discontinuous vector field in \( S \) but also an ambiguous one as there are six choices in every triangle. The construction of a covering surface replaces the need to handle the matchings \( r_{ij} \) in Eqn. (2). They are fully determined by \( S' \), i.e. the matchings are implicitly given by the choice of how the six copies of adjacent triangles were identified in \( S' \). The problem is reduced to finding a parameterization \((u', v')\) on \( S' \) which is continuous up to a translational offset by \( t_{ij} \) in adjacent triangles.

Lattice of Hexagonal Parameterizations: Due to the symmetry of the covering surface and the symmetric behavior of the algorithm, the resulting texture images on different copies of each triangle are congruent, i.e. for two points \( p_0, p_1 \) in the same fiber but on different layers of \( S' \), the parameter values are related by

\[
(u'(p_1), v'(p_1)) = R_0^6(u'(p_0), v'(p_0)) \oplus G_0.
\]

where \( G_0 \) is the set of possible values for translational discontinuities along seams and singularities. Because of this property, we can now take the parameterization of \( S' \) and project it back onto \( S \) by just taking the function value at any of the six layers. The choice of layer is unimportant because of Equation 4 and the symmetry of the texture image (see Figure 9).

Note that \( S' \) is a Riemannian surface with branch points at those positions where the original 6-RoSy field has
-processing step that is similar to the approach of Bommes T-junctions in remeshing. We correct this with a postprocessing step that is similar to the approach of Bommes T-junctions in remeshing. We correct this with a post-

decorating along these curves and thus cracks in surface tiling and distortions in the parameter lines, which can lead to mismatches between parameter lines along these curves and thus cracks in surface tiling and T-junctions in remeshing. We correct this with a postprocessing step that is similar to the approach of Bommes et al. [14] which seeks to reduce this mismatch along the cut curves. Note that the problem of finding the $L_2$-smallest functions $u'_a$, $v'_a$, such that all $t_{ij}$’s of the final parameterization are in $G_6$ is NP hard, since it is equivalent to minimizing a quadratic function on a given lattice (also called the closest vector problem). If the closest vector problem could be solved exactly in polynomial time, then the result would be independent of the chosen cut graph. Figure 11 shows the hexagonal parameterization of two minimal surfaces using our technique.

Note, that this is only a heuristic for the problem of finding the $L_2$-smallest functions $u'_a$, $v'_a$, such that all gaps of the final parameterization are in $G_6$. In general, this problem is NP hard, since it is equivalent to minimizing a quadratic function on a given lattice (also called the closest vector problem). In all our tests, the parameterization generated with the heuristic do not differ very much from the optimal solution. Furthermore, the heuristic seems to give more accurate results if the geometric length of the cut paths are smaller. This is the reason, why the shortest possible cut graph is a good choice. If we would solve the closest vector problem exactly, then the result would be independent of the chosen cut graph. Figure 11 shows the hexagonal parameterization of two minimal surfaces using our technique.

5 Results and Applications

In this section, we apply our hexagonal parameterization to two graphics applications: pattern synthesis on surfaces, and triangular remeshing.

5.1 Pattern Synthesis on Surfaces

Example-based texture and geometry synthesis on surfaces has received much attention from the Graphics community in recent years. We refer to [42] for a complete survey. Here we will refer to most relevant work.

Wei and Levoy [19] are the first to point out that some textures exhibit symmetries that make $N$-RoSy fields of $N > 1$ more suitable for the specification of orientation. Liu et al. [43] propose techniques for the analysis, manipulation, and synthesis of near-regular textures (i.e. very

Note that $G$ because of the repeating structure of the texture image. Is a discontinuous map, the discontinuities are not visible near-regular

Thus, the texture image is invariant under translations by any offset vector $t \in G_6$ and by rotation of about multiples of $\pi/3$ around any point in $G_6$. While a hex parameterization is a discontinuous map, the discontinuities are not visible because of the repeating structure of the texture image. Note that $G_6$ is exactly the Eisenstein integer lattice, which is different from the Gauss integer lattice that applies to quadrangular global parameterizations.

**Parameterization Generation:** The energy minimization problem in Equation 3 is solved using a conjugate gradient solver. The surface is then cut open to a topological disk and the resulting vector fields are integrated yielding the parameter functions $u'$ and $v'$. This parameterization is continuous everywhere ($t_{ij} = 0$ at all edges) except along the cuttings, where the parameter lines might not fit together.

To cut the surface open, we compute the shortest homotopy generators as done in [38]. In order to keep the covering symmetric, it is important that the cuts are identical in all layers, i.e. we first cut the original surface $S$ and then lift these cuts onto the covering $S'$. We also need to connect all singularities with the cut graph, since they can be seen as infinitesimal small holes in the surface. For this purpose, the method was adapted allowing also surface boundary and singularities by [39].

Up to this point the parameterization that we have obtained do not necessarily satisfy that $t_{ij} \in G_6$ along the cut curves, which can lead to mismatches between parameter lines along these curves and thus cracks in surface tiling and T-junctions in remeshing. We correct this with a postprocessing step that is similar to the approach of Bommes and Wei et al. [14] which seeks to reduce this mismatch along the cut curves. Note that the problem of finding the $L_2$-smallest functions $u'_a$, $v'_a$, such that all $t_{ij}$’s of the final parameterization are in $G_6$ is NP hard, since it is equivalent to minimizing a quadratic function on a given lattice (also called the closest vector problem). If the closest vector problem could be solved exactly in polynomial time, then the result would be independent of the chosen cut graph. Figure 11 shows the hexagonal parameterization of two minimal surfaces using our technique.

5 Results and Applications

In this section, we apply our hexagonal parameterization to two graphics applications: pattern synthesis on surfaces, and triangular remeshing.

5.1 Pattern Synthesis on Surfaces

Example-based texture and geometry synthesis on surfaces has received much attention from the Graphics community in recent years. We refer to [42] for a complete survey. Here we will refer to most relevant work.

Wei and Levoy [19] are the first to point out that some textures exhibit symmetries that make $N$-RoSy fields of $N > 1$ more suitable for the specification of orientation. Liu et al. [43] propose techniques for the analysis, manipulation, and synthesis of near-regular textures (i.e. very

Note that $G$ because of the repeating structure of the texture image. Is a discontinuous map, the discontinuities are not visible near-regular

Thus, the texture image is invariant under translations by any offset vector $t \in G_6$ and by rotation of about multiples of $\pi/3$ around any point in $G_6$. While a hex parameterization is a discontinuous map, the discontinuities are not visible because of the repeating structure of the texture image. Note that $G_6$ is exactly the Eisenstein integer lattice, which is different from the Gauss integer lattice that applies to quadrangular global parameterizations.

**Parameterization Generation:** The energy minimization problem in Equation 3 is solved using a conjugate gradient solver. The surface is then cut open to a topological disk and the resulting vector fields are integrated yielding the parameter functions $u'$ and $v'$. This parameterization is continuous everywhere ($t_{ij} = 0$ at all edges) except along the cuttings, where the parameter lines might not fit together.

To cut the surface open, we compute the shortest homotopy generators as done in [38]. In order to keep the covering symmetric, it is important that the cuts are identical in all layers, i.e. we first cut the original surface $S$ and then lift these cuts onto the covering $S'$. We also need to connect all singularities with the cut graph, since they can be seen as infinitesimal small holes in the surface. For this purpose, the method was adapted allowing also surface boundary and singularities by [39].

Up to this point the parameterization that we have obtained do not necessarily satisfy that $t_{ij} \in G_6$ along the cut curves, which can lead to mismatches between parameter lines along these curves and thus cracks in surface tiling and T-junctions in remeshing. We correct this with a postprocessing step that is similar to the approach of Bommes and Wei et al. [14] which seeks to reduce this mismatch along the cut curves. Note that the problem of finding the $L_2$-smallest functions $u'_a$, $v'_a$, such that all $t_{ij}$’s of the final parameterization are in $G_6$ is NP hard, since it is equivalent to minimizing a quadratic function on a given lattice (also called the closest vector problem). If the closest vector problem could be solved exactly in polynomial time, then the result would be independent of the chosen cut graph. Figure 11 shows the hexagonal parameterization of two minimal surfaces using our technique.

5 Results and Applications

In this section, we apply our hexagonal parameterization to two graphics applications: pattern synthesis on surfaces, and triangular remeshing.

5.1 Pattern Synthesis on Surfaces

Example-based texture and geometry synthesis on surfaces has received much attention from the Graphics community in recent years. We refer to [42] for a complete survey. Here we will refer to most relevant work.

Wei and Levoy [19] are the first to point out that some textures exhibit symmetries that make $N$-RoSy fields of $N > 1$ more suitable for the specification of orientation. Liu et al. [43] propose techniques for the analysis, manipulation, and synthesis of near-regular textures (i.e. very

Note that $G$ because of the repeating structure of the texture image. Is a discontinuous map, the discontinuities are not visible near-regular

Thus, the texture image is invariant under translations by any offset vector $t \in G_6$ and by rotation of about multiples of $\pi/3$ around any point in $G_6$. While a hex parameterization is a discontinuous map, the discontinuities are not visible because of the repeating structure of the texture image. Note that $G_6$ is exactly the Eisenstein integer lattice, which is different from the Gauss integer lattice that applies to quadrangular global parameterizations.

**Parameterization Generation:** The energy minimization problem in Equation 3 is solved using a conjugate gradient solver. The surface is then cut open to a topological disk and the resulting vector fields are integrated yielding the parameter functions $u'$ and $v'$. This parameterization is continuous everywhere ($t_{ij} = 0$ at all edges) except along the cuttings, where the parameter lines might not fit together.

To cut the surface open, we compute the shortest homotopy generators as done in [38]. In order to keep the covering symmetric, it is important that the cuts are identical in all layers, i.e. we first cut the original surface $S$ and then lift these cuts onto the covering $S'$. We also need to connect all singularities with the cut graph, since they can be seen as infinitesimal small holes in the surface. For this purpose, the method was adapted allowing also surface boundary and singularities by [39].

Up to this point the parameterization that we have obtained do not necessarily satisfy that $t_{ij} \in G_6$ along the cut curves, which can lead to mismatches between parameter lines along these curves and thus cracks in surface tiling and T-junctions in remeshing. We correct this with a postprocessing step that is similar to the approach of Bommes and Wei et al. [14] which seeks to reduce this mismatch along the cut curves. Note that the problem of finding the $L_2$-smallest functions $u'_a$, $v'_a$, such that all $t_{ij}$’s of the final parameterization are in $G_6$ is NP hard, since it is equivalent to minimizing a quadratic function on a given lattice (also called the closest vector problem). If the closest vector problem could be solved exactly in polynomial time, then the result would be independent of the chosen cut graph. Figure 11 shows the hexagonal parameterization of two minimal surfaces using our technique.
structured textures with repeating patterns) in the plane. Kaplan and Salesin [44] address the design of Islamic star patterns in the plane. There has been some recent work in constructing circle patterns from a triangular mesh for architectural models [45].

Generating regular patterns on a surface can be greatly facilitated given an appropriate global parameterization. Given a regular hexagonal texture or geometry pattern, it is simply tiled in the parameter-space of the mesh and the texture should stitch (relatively) seamlessly everywhere (Figure 12). For example, to achieve circle packing for architectural patterns, our hexagonal parameterization allows nice hexagonal patterns to be generated from a surface, which can be used as input to such algorithms as shown in Figure 12 (right). Our method provides necessary smoothness and feature alignment, thus leading to a high-quality model, even in the case of relatively high geometric and topological complexity. Figure 3 (b, c) provides some additional examples in which regular hexagonal texture and geometry patterns are placed on the dragon.

We also comment that quadrangular global parameterizations can be used to synthesis patterns of 4-RoSys on surfaces. Figure 8 provides one such example. To the best of our knowledge this realization has not been published. Notice that our field generation algorithm can also automatically generate geometry-aware 4-RoSy fields, which lead to coherent synthesized patterns that align with surface features.

5.2 Triangular Remeshing

There has been much work in triangular remeshing. To review all past work is beyond the scope of this paper. We refer the reader to [46] for a complete survey of triangular remeshing literature, and review only the most relevant work here. Common methods of mesh triangulation are typically based on either a parameterization [47], [40], [48], [49], local optimization methods [50], [51], [52], or Delaunay triangulations and centroidal Voronoi tessellations [53], [41].

The focus of triangular remeshing is on shape preservation, good triangle aspect ratio, feature-aware triangle sizing, and control of irregular vertices (valence not equal to six). These object functions are often conflicting with one another, and the output mesh is a result of compromise among these factors. For example, many parameterization-based methods suffer from artifacts in the triangulation at the locations of the chart boundaries (though this problem can be alleviated by using a global parameterization as in [49]). Direct and local optimization methods suffer from a lack of global control over the structure of the triangulation such as the location and number of irregular vertices.

In this paper, we perform triangular remeshing using a hexagonal global parameterization derived a shape-aware 6-RoSy field. There are a number of benefits to this. First, such an approach can lead to overall better aspect ratio for triangles in the remesh (equilateral). Second, the number of irregular vertices can be reduced and their locations can be controlled as these vertices correspond exactly to the set of singularities in the 6-RoSy field. Third, we have incorporated the ability to match the orientations of the RoSy field based on natural anisotropy in the surfaces. Fourth, the size of the triangles can be controlled through a scalar sizing function (Figure 14).

Figure 13 compares the results of the foot and Venus models using our method with that of [40], which is based on Delaunay triangulations, and [41], which is based on restricted Voronoi diagrams. Table 1 provides the quality statistics of all the test models used in this paper as well as the comparison to [40], [41]. Notice that all three methods...
Fig. 13. We compare the outputs of our method (right) to those of Alliez et al. [40] (left) and Yan et al. [41] (middle) with a Foot model and Venus. Note the reduced number of irregular vertices in flat regions of the Foot, and increased feature alignment (around the nose and brow) on the Venus. The histograms show the occurring inner angles (on the X-axis from 0 to 120 degrees).

Fig. 14. We demonstrate our capability to have variable sizes of triangles (right) given a scalar sizing function (left).

capture the underlying geometry well (comparable Hausdorff distances to the original input mesh) but our method tends to have the fewest irregular vertices among all three methods and smaller variations in angles of the triangles than [40]. Note that having relatively few irregular vertices is a direct result of our automatic singularity clustering in the field generation (Section 3) while achieving good triangle aspect ratios is due to the nature of the hexagonal parameterization. In addition, our method tends to produce edge directions that better align with the features in the mesh, such as along Venus’ nose ridge than [40]. Additional remeshing results can be found in Figure 3.

5.3 Performance

The amount of time to automatically generate a geometry-aware 6-RoSy field is on average 40 seconds for a model of 40K triangles, measured on a PC with a 2.4GHz CPU. The time to generate the parameterization is approximately 120 seconds per model, measured on a PC with 2.13GHz CPU. The running time of both stages are impacted by the mesh size as well as the number of singularities in the RoSy field.
appropriate mathematical representations that handle more sophisticated types of wallpaper textures which may contain reflections and gliding reflections. We will also investigate means to improve the quality of parameterization by better balancing between the topological complexity (number of singularities) and distortion in the parameterization.

### References


---

**TABLE 1**

Mesh quality of all test models used in this paper. We have also included comparisons with two models (foot, Venus) remeshed using the methods of Alliez et al. [40] and Yan et al. [41], respectively. The mesh quality is measured in terms of the Hausdorff distance (in % of the bounding box), the minimum, maximum, and standard deviation (SD) of the angles in the mesh measured in degrees, as well as the number irregular vertices. Notice that our method produces similar approximation errors (Hausdorff distance), but leads to better overall aspect ratios of triangles in the mesh (larger minimum angle, smaller maximum angle, and small standard deviation of angles) than [40] as well as fewest irregular vertices among all three methods.

<table>
<thead>
<tr>
<th>model name</th>
<th>Hausdorff distance</th>
<th>min angle</th>
<th>max angle</th>
<th>SD angle</th>
<th># irregular vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>foot [40]</td>
<td>0.3373</td>
<td>2.82</td>
<td>173.88</td>
<td>11.92</td>
<td>146</td>
</tr>
<tr>
<td>foot [41]</td>
<td>0.0094</td>
<td>26.92</td>
<td>115.85</td>
<td>7.40</td>
<td>32.82</td>
</tr>
<tr>
<td>foot</td>
<td>0.0129</td>
<td>22.65</td>
<td>125.09</td>
<td>5.11</td>
<td>13</td>
</tr>
<tr>
<td>Venus [40]</td>
<td>0.1005</td>
<td>0.42</td>
<td>178.99</td>
<td>17.48</td>
<td>38</td>
</tr>
<tr>
<td>Venus [41]</td>
<td>0.0439</td>
<td>19.89</td>
<td>121.13</td>
<td>10.37</td>
<td>1449</td>
</tr>
<tr>
<td>Venus</td>
<td>0.0543</td>
<td>24.87</td>
<td>114.80</td>
<td>6.84</td>
<td>36</td>
</tr>
<tr>
<td>Max Planck</td>
<td>0.00263</td>
<td>12.44</td>
<td>145.79</td>
<td>5.00</td>
<td>44</td>
</tr>
<tr>
<td>bunny</td>
<td>0.00751</td>
<td>18.03</td>
<td>138.54</td>
<td>7.54</td>
<td>65</td>
</tr>
<tr>
<td>teline</td>
<td>0.02695</td>
<td>3.75</td>
<td>167.34</td>
<td>11.09</td>
<td>121</td>
</tr>
<tr>
<td>dragon</td>
<td>0.00762</td>
<td>4.80</td>
<td>151.88</td>
<td>9.10</td>
<td>181</td>
</tr>
<tr>
<td>Buddha</td>
<td>0.02338</td>
<td>1.89</td>
<td>175.11</td>
<td>12.12</td>
<td>123</td>
</tr>
</tbody>
</table>

---

**6 CONCLUSIONS**

In this paper we have introduced a novel type of global parameterizations which we refer to as hexagonal global parameterizations. We demonstrate the uses of such parameterization with applications in pattern synthesis on surfaces and triangular remeshing.

We provide a two-step pipeline to generate a shape-aware hexagonal global parameterization. First, a 6-RoSy field is automatically generated based on the surface geometry. Second, the 6-RoSy field is used to generate the hexagonal global parameterization based on the idea of covering spaces. As part of our field generation method, we allow automatic constraint selection by applying the trace-and-deviator decomposition to the curvature tensor. We have also provided the capability to automatically cluster nearby singularities into higher-order ones, thus reducing topological noise. While these techniques are demonstrated for the 6-RoSy fields, they are generic and can apply to 4-RoSy fields with relatively few modifications.

There are a number of possible future research directions. First, we wish to study objects that are close to N-RoSy, which we refer to as near-regular RoSyS. In these objects the N member vectors do not have identical magnitude nor even angular spacings. Such objects can allow more flexibility in both quadrangular and triangular remeshing. Second, five-fold symmetry appears in many natural objects such as flowers. We wish to pursue graphics applications that deal with 5-RoSyS. It is also our intent to investigate...